Signal Processing by Switching Markov State-Space Models: Estimation of the State of Charge of an Electric Battery

Jana Kalawoun, Patrick Pamphile

Signal Processing by Switching Markov State-Space Models: Estimation of the State of Charge of an Electric Battery

Jana Kalawoun[a,*], Patrick Pamphile[b]

[a] *CEA, LIST, Laboratoire d'Analyse de Données et Intelligence des Systèmes, 91191 Gif-sur-Yvette, France*
[b] *INRIA Université Paris Sud CNRS UMR8628 91405 Orsay, France*

Abstract

Switching Markov State-Space Models (SMSSM) are linear models whose parameters randomly change over time according to a finite discrete Markov chain. This generalization allows, for instance, to dealing with systems which are locally linear. However, it can be difficult to implement SMSSM on real-world applications. In this paper we present techniques and methods to solve the four basic problems of SMSSM implementation, namely the identifiability, the model parameters inference, the order selection and the online state inference. As an illustration, we consider the problem of estimating the State of Charge (SoC) of an electric battery. For this purpose, we develop a new SoC model, implemented with a SMSSM, and show its ability to accurately estimate the SoC of the battery of an electric vehicle under different usage conditions.

Keywords: Switching Markov state-space model, EM algorithm, Particle filter, Order selection, Electric vehicle, State of charge

1. Introduction

State-space models are widely used in several fields of applied science such as signal processing, economics and bioinformatics [1, 2]. They are defined

*Corresponding author
Email addresses: jana.kalawoun@cea.fr (Jana Kalawoun), patrick.pamphile@u-psud.fr (Patrick Pamphile)

by a "state equation" describing the dynamics of the unobservable state of interest of the system X_t, and an "observation equation" specifying how the distribution of the observations Y_t is influenced by X_t. Switching Markov State-Space Models (SMSSM) are used when the underlying dynamics of the system are expected to switch at unknown dates, depending on uncontrolled internal and external conditions (see for instance [3]). These switches in the system behavior are indexed by a discrete hidden Markov chain S_t.

We distinguish four basic problems that must be solved for a SMSSM to be useful in real-world applications. These problems are as follows:

1. **Identifiability:** A model is said identifiable when there is only one set of parameters which leads to a given input-output behavior. It is well-known that a SMSSM is not identifiable. Thus, the parameters inference can be misleading. Therefore given an observation sequence $y_{0:T} = \{y_0, \ldots, y_T\}$ and an input sequence $u_{0:T}$, which sensible constraints can be imposed to ensure the identifiability of the model?
2. **Off-line model parameters inference:** Given an observation sequence $y_{0:T}$, how to infer the model parameters Θ maximizing the likelihood $p_\Theta(y_{0:T})$?
3. **Order selection:** A hidden Markov states models a specific regime of the underlying dynamics of the system. Thus, given an observation sequence $y_{0:T}$, how to identify the proper number of hidden Markov states?
4. **Online state inference:** Given an observation y_t and the model parameters Θ, how the corresponding state of interest x_t can be online estimated?

This paper is intended to provide an overview of the basic problems of SMSSMs, and practical details on methods of implementation of these models in real-world applications. The identifiability of parameters of a SMSSM is essential to obtain a relevant inference of the model parameters. Notwithstanding its importance, the identifiability issue of SMSSMs has not been addressed yet. In this paper, we prove that the identifiability of SMSSM can be achieved by imposing straightforward and natural constraints on the parameters. The EM algorithm is particularly useful to off-line estimate the parameters, given that the state of interest X_t and the Markov state S_T are non-observable (see [3] for an excellent review of SMSSM parameters estimation). For a large training dataset, we will show that the EM algorithm must be approximated using Monte Carlo methods. In practice, the number

of hidden Markov states is unknown, and a proper choice of this number is crucial to explain what factors influence the system dynamics, and accurately estimate the state of interest x_t. Once the number of Markov states has been identified and the corresponding set of parameters estimated, the state of interest x_t is then accurately estimated online using a particle filter. Moreover, the numerical difficulties in implementing of these methods, mainly the calibration of the Monte Carlo EM algorithm, are thoroughly studied.

As illustration, we consider a challenging signal processing issue, namely the estimation of the State of Charge (SoC) of an electric battery. Nowadays, to achieve better fuel economy and reduce toxic emissions, more and more vehicles are powered with an electric motor. Similarly to the fuel gauge in a standard vehicle, the SoC indicates its available energy. Thus, the estimation of SoC of the battery is an essential information to guarantee the vehicle autonomy, as well as its safe utilization. Indeed, the battery dynamics changes according to uncontrolled internal and external usage conditions. Various studies have been conducted to take into account these changes [4, 5, 6], but the proposed solutions remain contestable. In this paper, we propose a new model for the SoC estimation using a SMSSM, and test it on real-world electric vehicle data.

The organization of this paper is as follows. In Section 2, we present the signal model and the assumptions used. In Section 3, we address the identifiability problem of a SMSSM. In Section 4, an off-line learning of the unknown parameters is developed using a Monte Carlo approximation of the EM algorithm. In Section 5, the choice of the optimal number of Markov states κ is addressed using different model selection criteria. In Section 6, we show that an accurate estimation of the state of interest x_t requires a prohibitive computational cost, and present several suboptimal estimation algorithms for solving this problem. In Section 7, we discuss the numerical difficulties in implementing SMSSMs based on simulated data. In Section 8, we develop an electric battery state of charge estimator, implemented using SMSSMs, and show how it takes into account the random changes of the dynamics of the battery and accurately estimates the state of charge for different usage conditions. A discussion section ends the paper.

2. Switching Markov State-Space Models

Consider an observable random process $Y_{0:T} = \{Y_0, \ldots, Y_T\}$ where the realizations are denoted by $\{y_0, \ldots, y_T\}$. Let S_t denote a discrete irreducible

and aperiodic Markov chain on $\{1, \ldots, \kappa\}$, with initial distribution Π and transition matrix P. The distribution of Y_t is allowed to depend on an unobservable latent variable X_t, which represents the continuous state of interest, through the observation equation

$$Y_t = C(S_t)X_t + D(S_t)u_t + \varepsilon_t, \tag{1}$$

for $t = 0, 1, \ldots, T$, where $\varepsilon_t \sim \mathcal{N}(0, \sigma_Y^2(S_t))$ models the sensor error. The known exogenous input u_t belongs to $\mathbb{R}^n$. For simplicity sake, we consider that X_t and $Y_t \in \mathbb{R}$. The description of the model is completed by the transition equation describing the evolution of X_t

$$X_t = A(S_t)X_{t-1} + B(S_t)u_t + \omega_t, \tag{2}$$

where $\omega_t \sim \mathcal{N}(0, \sigma_X^2(S_t))$ models the fluctuation of X_t. The Gaussian white noises ω_t and ε_t are assumed to be independent conditionally on S_t. The augmented state $\{X_t, S_t\}$ is a Markov chain, verifying in addition

$$p_\Theta(s_t, x_t \mid s_{t-1}, x_{t-1}) = p_\Theta(s_t \mid s_{t-1}) p_\Theta(x_t \mid x_{t-1}, s_t), \tag{3}$$

where Θ is the set of model parameters

$$\Theta = \{P, \Gamma = (A(s), B(s), \sigma_X(s), C(s), D(s), \sigma_Y(s))\}, \tag{4}$$

for $s = 1, \ldots, \kappa$. Indeed, a SMSSM can be viewed as a Hidden Markov Model (HMM) with two latent states (i.e., X_t and S_t). The observations (Y_t) are assumed to be independent given $\{S_t, X_t\}$. It is also assumed that x_0 is fixed, the initial distribution Π is known, and the distributions $p_\Theta(y_t \mid x_t, s_t)$ and $p_\Theta(x_t \mid x_{t-1}, s_t)$ are Gaussian with parameters deduced respectively from (1) and (2).

The identifiability issue for the SMSSM defined above is addressed in the next section. It is essential for a relevant parameters identification.

3. Identifiability of a SMSSM

A model is said "identifiable" if there exists one set of parameters Θ which leads to a given input-output relation. This can be formulated as follows

$$p_\Theta(y_{0:T}) = p_{\Theta^*}(y_{0:T}) \quad \Rightarrow \quad \Theta = \Theta^*. \tag{5}$$

Moreover, a subset of parameters $F \subset \Theta$ is said "globally structurally (g.s.) identifiable" when $p(y_{0:T}|\Theta^*) = p(y_{0:T}|\Theta)$ implies $F^* = F$. However, it is well-known that a linear state-space model [7], and even more a SMSSM, suffers from identifiability problems. Indeed, let us consider that X_t^* and X_t are related by a linear transformation $X_t^* = H(s_t)X_t$. We obtain an equivalent model with the same observation sequence

$$\left\{ \begin{array}{rcccl} A^*(s) & = & H(s) & \cdot & A(s) \cdot H^{-1}(s) \\ B^*(s) & = & H(s) & \cdot & B(s) \\ C^*(s) & = & C(s) & \cdot & H^{-1}(s) \\ {\sigma_X^2}^*(s) & = & H(s) & \cdot & \sigma_X^2(s) \cdot H(s), \end{array} \right. \tag{6}$$

where $H(s) \in \mathbb{R}$. It has to be noted that $D(s)$ and $\sigma_Y(s)$ are always g.s. identifiable. They are invariant under any linear transformation $H(s)$. Also, $A(s)$ is g.s. identifiable since it belongs to $\mathbb{R}$.

To ensure the identifiability of the model, constraints could be imposed on its parameters. It is noteworthy that the Markov states S_t can be relabeled without changing the distribution of the observations [8]. Thus, the identifiability of the SMSSM is considered up to state switching.

3.1. Case of LGSSM

As a first step, we address the identifiability of a Linear Gaussian State-Space Model (i.e., SMSSM with $\kappa = 1$). The following prior information is considered at $t_0 = 0$

$$y_0 = Cx_0 + Du_0, \tag{7}$$

with $x_0 \cdot u_0 \neq 0_{n \times 1}$. Any equivalent model should have the same constraint

$$y_0 = C^*x_0 + D^*u_0 = CH^{-1}x_0 + Du_0. \tag{8}$$

Consequently, under (7) and (8), the only solution of (6) is $H = 1$. Thus, the parameters of a SMSSM with $\kappa = 1$ are g.s. identifiable under (7).

3.2. Case of known sequence Markov states

It is assumed that at $t_0 = 0$

$$y_0 = C(s_0)x_0 + D(s_0)u_0, \tag{9}$$

for any hidden state $s_0 = 1, \ldots, \kappa$. When the Markov sequence $s_{0:T}$ is known, the model can be transformed into κ LGSSMs with an appropriate sampling

time. Accordingly, the identifiability results of the previous section can be extended, and the set Γ of parameters of these κ LGSSMs is g.s. identifiable under constraints (9). This can be formulated by the following relation:

$$p_\Gamma(y_{0:T}|s_{0:T}) = p_{\Gamma^*}(y_{0:T}|s_{0:T}) \quad \Rightarrow \quad \Gamma = \Gamma^*. \tag{10}$$

3.3. Case of unknown sequence of Markov states

When the Markov sequence $s_{0:T}$ is unknown, the marginal likelihood of the SMSSM (1)-(2) is given by

$$p_\Theta(y_{0:T}) = \sum_{s_{0:T}\in\mathcal{S}} p_P(s_{0:T}) \cdot p_\Gamma(y_{0:T}|s_{0:T}), \tag{11}$$

where $\mathcal{S} = \{1, \ldots, \kappa\}^{T+1}$ and $p_\Gamma(y_{0:T}|s_{0:T})$ is a Gaussian distribution whose parameters are recursively calculated using a Kalman filter. Thus $p_\Theta(y_{0:T})$ is a finite convex combination of Gaussian distributions. Yakowitz and Spragins [9] prove that a family of finite mixture distributions is identifiable, iff the members of the underlying distribution family are linearly independent over the field of real numbers. Based on this theorem, they then prove that a T-dimensional Gaussian distribution $p_\Gamma(y_{0:T}|s_{0:T})$ generates identifiable finite mixtures. Hence, we have

$$\begin{aligned} p_\Theta(y_{0:T}) = p_{\Theta^*}(y_{0:T}) \Rightarrow & 1) p_\Gamma(y_{0:T}|s_{0:T}) = p_{\Gamma^*}(y_{0:T}|s_{0:T}) \\ & 2) p_P(s_{0:T}) = p_{P^*}(s_{0:T}). \end{aligned} \tag{12}$$

Under the constraints (9), the first equation implies that $\Gamma = \Gamma^*$. Since s_t is an irreducible aperiodic Markov chain, thus has a unique stationary distribution, the second equation implies that $P = P^*$ cf. Lemma 2 in [10]. As a result, we have the following proposition.

Proposition 1. *The parameters of the SMSSM (1)-(2) are g.s. identifiable if the following constraints are verified*

1. *A prior information at $t_0 = 0$ is available:*

$$y_0 = C(s)x_0 + D(s)u_0 \text{ with } s = 1, \ldots, \kappa \text{ and } x_0 \cdot u_0 \neq 0_{n\times 1}, \tag{13}$$

2. $\forall (i, j)$, $P(i, j) \neq 0$.

Condition 2 implies that the finite hidden Markov chain is irreducible and aperiodic. In the next section, under the above constraints, we attempt to

estimate the model parameters Θ maximizing the probability of the observation sequence.

4. Off-line learning of model parameters

Let us consider a learning dataset $\{y_{0:T}, u_{0:T}\}$ where $y_{0:T}$ is observed and $u_{0:T}$ is an input. Both $x_{0:T}$ and $s_{0:T}$ are unknown. In the following, an off-line learning of unknown parameter Θ using the Maximum Likelihood (ML) inference is proposed. The original ML estimation problem can be formulated as follows

$$\widehat{\Theta}_{ML} = \arg\max_{\Theta} \log p_{\Theta}(y_{0:T}). \tag{14}$$

For SMSSM, the marginal likelihood is given by

$$p_{\Theta}(y_{0:T}) = \sum_{s_{0:T}} \int_{x_{0:T}} p_{\Theta}(x_{0:T}, s_{0:T}, y_{0:T}) dx_{0:T}, \tag{15}$$

where $p_{\Theta}(x_{0:T}, s_{0:T}, y_{0:T})$ is the complete-likelihood given by

$$\begin{aligned} p_{\Theta}(x_{0:T}, s_{0:T}, y_{0:T}) =& p_{\Theta}(s_0) p_{\Theta}(x_0 \mid s_0) p_{\Theta}(y_0 \mid s_0, x_0) \\ & \cdot \prod_{t=1}^{T} p_{\Theta}(s_t \mid s_{t-1}) \cdot p_{\Theta}(x_t \mid x_{t-1}, s_t) \cdot p_{\Theta}(y_t \mid x_t, s_t). \end{aligned} \tag{16}$$

It is clear that a direct evaluation of (15) is not analytically tractable as it involves the summation over up to κ^{T+1} Markov sequences. Therefore the EM algorithm is used to iteratively compute the maximum likelihood estimate $\widehat{\Theta}_{ML}$. This algorithm is particularly useful in incomplete-data problems [11]. The EM iteration alternates between an expectation step (E-step), which calculates the conditional expectation of the log-likelihood for complete data using the current estimated parameters, and a maximization step (M-step), which computes the parameters maximizing the conditional expectation found on the E-step. These estimated parameters are then used in the E-step of the next iteration.

4.1. EM algorithm for SMSSM

Instead of $\log p_\Theta(y_{0:T})$, the EM algorithm considers the conditional expectation of the complete log-likelihood

$$\mathcal{Q}(\Theta, \Theta') = \mathbb{E}_{Y_{0:T},\Theta'} \left[\log p_\Theta \left(X_{0:T}, S_{0:T}, Y_{0:T}\right)\right]. \tag{17}$$

Given an initial value of Θ', the ML estimator is then iteratively approached by Θ which maximizes $\mathcal{Q}(\Theta, \Theta')$ [11].

4.1.1. Expectation Step

Based on the interaction (3) between the two latent variables S_t and X_t, the conditional expectation $\mathcal{Q}(\Theta, \Theta')$ can be written as follows

$$\mathcal{Q}(\Theta, \Theta') = \sum_{s_{0:T}} \{p_{\Theta'}(s_{0:T} \mid y_{0:T}) \mathbb{E}_{S_{0:T},Y_{0:T},\Theta'}[\log p_\Theta(X_{0:T}, S_{0:T}, Y_{0:T})]\}, \tag{18}$$

where the conditional expectation in (18) is evaluated using a Kalman filter.

4.1.2. Maximization Step

The maximization of $\mathcal{Q}(\Theta, \Theta')$ w.r.t Θ must be conducted under the constraint on the transition matrix $\sum_j P(i,j) = 1$ and the constraints (13) ensuring the identifiability of the parameters. The Lagrangian associated to these constraints is

$$\mathcal{L}(\Theta,\, \lambda,\, \mu) = \mathcal{Q}(\Theta, \Theta') + \sum_{i=1}^{\kappa} \lambda_i [1 - \sum_j P(i,j)] + \mu_i [y_0 - C(i)x_0 - D(i)u_0], \tag{19}$$

where λ_i and μ_i, $1 \leq i \leq \kappa$, are the Lagrangian multipliers. The transition matrix P is estimated by solving $\partial \mathcal{L}(\Theta, \lambda, \mu)/\partial P = 0$

$$\widehat{P}(i,j) = \frac{\sum\limits_{\{t : s_{t-1} = i\,,\, s_t = j\}} p_{\Theta'}(s_{t-1} = i\,,\, s_t = j \,|\, y_{0:T})}{\sum\limits_{\{t : s_{t-1} = i\}} p_{\Theta'}(s_{t-1} = i \,|\, y_{0:T})}. \tag{20}$$

Canceling $\partial \mathcal{L}(\Theta, \lambda, \mu)/\partial \Gamma$ leads to the following system of $(4{+}2n)\kappa$ equations

$$\sum_{t : s_t = i} \left[A(i)\widehat{x}_{t-1,t-1|T} + B(i)\widehat{x}_{t-1|T} u_t - \widehat{x}_{t-1,t|T}\right] \Pi_{t|T}(i) = 0, \tag{21}$$

$$\sum_{t:s_t=i} \left[A(i)I_n u_t \widehat{x}_{t-1|T} + I_n u_t u_t' B'(i) - I_n u_t x_t\right] \Pi_{t|T}(i) = 0, \tag{22}$$

$$\sum_{t:s_t=i} \left[C(i)\widehat{x}_{t,t|T} + D(s)\widehat{x}_{t|T} u_t - \widehat{x}_{t|T} y_t\right] \Pi_{t|T}(i) + \lambda_i \sigma_Y(i) x_0 = 0, \tag{23}$$

$$\sum_{t:s_t=i} \left[C(i)I_n u_t \widehat{x}_{t|T} + I_n u_t u_t' D'(i) - I_n u_t y_t\right] \Pi_{t|T}(i) + \lambda_i \sigma_Y(i) u_0 = 0, \tag{24}$$

$$\sum_{t:s_t=i} \left[\sigma_X^2(i) - \widehat{x}_{t-1,t-1|T} - A^2(i)\widehat{x}_{t,t|T} - B(i)u_t B(i)u_t + 2A(i)\widehat{x}_{t,t-1|T} \right.$$
$$\left. + 2\widehat{x}_{t|T} B(i)u_t - 2A(i)\widehat{x}_{t|T} B(i)u_t\right] \Pi_{t|T}(i) = 0, \tag{25}$$

$$\sum_{t:s_t=i} \left[\sigma_Y^2(i) - y_t^2 - C^2(i)\widehat{x}_{t,t|T} - D(i)u_t D(i)u_t + 2y_t C(i)x_t + 2y_t D(i)u_t \right.$$
$$\left. - 2C(i)x_t D(i)u_t\right] \Pi_{t|T}(i) = 0, \tag{26}$$

where I_n is the identity matrix of dimension n, $\widehat{x}_{t|T} = \mathbb{E}[X_t \mid s_{0:T}, y_{0:T}, \Theta']$, $\widehat{x}_{t,r|T} = \mathbb{E}[X_t \cdot X_r \mid s_{0:T}, y_{0:T}, \Theta']$ and $\Pi_{t|T}(i) = p_{\Theta'}(s_t = i \mid y_{0:T})$.

However, an exact computation of this system (21)-(26) needs to perform summations over up to κ^{T+1} values of $s_{0:T}$. Even for modest values of T, this would be too expensive.

4.2. Monte Carlo approximation of the EM algorithm for SMSSM

To overcome this computational difficulty, we resort to the Monte Carlo method to numerically approximate the EM algorithm [12]. More precisely, a set of N independent "particles" $s_{0:T}^i$ is simulated from $p_{\Theta'}(s_{0:T} \mid y_{0:T})$, and then $\partial \mathcal{Q}(\Theta, \Theta')/\partial\Theta$ is estimated by

$$\frac{1}{N} \sum_{i=1}^{N} \frac{\partial}{\partial \Theta} \mathbb{E}_{s_{0:T}^i, Y_{0:T}, \Theta'}[\log p_\Theta(X_{0:T}, s_{0:T}^i, Y_{0:T})]. \tag{27}$$

Unfortunately, it is difficult to simulate directly from the "target" distribution $p_{\Theta'}(s_{0:T} \mid y_{0:T})$. Therefore, we use the well-known "importance sampling" method. The basic idea of this method is to sample from an "instrumental distribution" $q_{\Theta'}(s_{0:T} \mid y_{0:T})$ from which it is easy to obtain samples, and then to introduce "importance weights" to obtain an unbiased estimator. These importance weights are given by

$$w_T^i = w(s_{0:T}^i) = \frac{p_{\Theta'}(s_{0:T} \mid y_{0:T})}{q_{\Theta'}(s_{0:T} \mid y_{0:T})}. \tag{28}$$

Accordingly, $\partial\mathcal{Q}(\Theta,\Theta')/\partial\Theta$ is estimated as follows

$$\sum_{i=1}^{N} \hat{w}_T^i \frac{\partial}{\partial\Theta} \mathbb{E}_{s_{0:T}^i, Y_{0:T}, \Theta'}[\log p_\Theta(X_{0:T}, s_{0:T}^i, Y_{0:T})], \tag{29}$$

where $\hat{w}_T^i$ denotes the normalized version of w_T^i. The strong law of large number implies that this estimate (29) converges to $\partial\mathcal{Q}(\Theta,\Theta')/\partial\Theta$ almost surely as N tends to infinity. The selecting of the instrumental distribution is discussed later in this section.

However, sampling from $q_{\Theta'}(s_{0:T} \mid y_{0:T})$ would have a computational complexity increasing at least linearly with T [12]. Hence, since S_t is a Markov chain, we naturally resort to the "sequential importance sampling" that admits a fixed computational complexity at each time step t.

4.2.1. Sequential Importance Sampling (SIS)

At time t, the instrumental distribution $q_{\Theta'}(s_{0:t} \mid y_{1:t})$ can be written as follows

$$q_{\Theta'}(s_{0:t} \mid y_{0:t}) = q_{\Theta'}(s_0) \prod_{k=1}^{t} q_{\Theta'}(s_t \mid s_{0:k-1},\, y_{0:t}). \tag{30}$$

Therefore at time $t+1$, the simulated sequences $s_{0:t}^i$ will be modified as the observation y_{t+1} is considered. In sequential sampling, our aim is to simulate s_t without modifying the previously simulated sequences $s_{0:t-1}^i$. This is possible if the instrumental distribution has the following form

$$q_{\Theta'}(s_{0:t} \mid y_{0:t}) = q_{\Theta'}(s_0) \prod_{k=1}^{t} q_{\Theta'}(s_t \mid s_{0:k-1},\, y_{0:k}). \tag{31}$$

Importance weight. The importance weights are recursively evaluated

$$w_t^i = w_{t-1}^i \cdot \tilde{w}_t^i, \tag{32}$$

where

$$\begin{aligned} \tilde{w}_t^i =& \frac{p_{\Theta'}(y_t \mid s_{0:t}, y_{0:t-1}) p_{\Theta'}(s_t \mid s_{0:t-1})}{p_{\Theta'}(y_t \mid y_{0:t-1}) q_{\Theta'}(s_t \mid s_{0:t-1}, y_{0:t})} \\ \propto& \frac{p_{\Theta'}(y_t \mid s_{0:t}, y_{0:t-1}) p_{\Theta'}(s_t \mid s_{0:t-1})}{q_{\Theta'}(s_t \mid s_{0:t-1}, y_{0:t})}. \end{aligned} \tag{33}$$

Sequential Importance Resampling. The entire path up to t of $p_{\Theta'}(s_{0:t} \mid y_{0:t})$ and $q_{\Theta'}(s_{0:t} \mid y_{0:t})$ could be far apart. Therefore after a few simulation iterations, a lot of importance weights could be very close to zero. To avoid this "degeneracy phenomenon", a resampling step is generally added to the SIS algorithm. It consists of discarding the particles $s^i_{0:t}$ with low importance weights and duplicating the ones with high $\hat{w}^i_t$. Here, we use the multinomial sampling procedure which is the most popular resampling procedure [13].

Choice of the instrumental distribution. The obvious condition for selecting the instrumental distribution is that $q_{\Theta'}(s_{0:T} \mid y_{0:T}) = 0$ implies $p_{\Theta'}(s_{0:T} \mid y_{0:T})\frac{\partial}{\partial \Theta} = 0$. This condition ensures that in equation (28) the denominator is not equal to zero.

Here we choose $q_{\Theta'}(s_t \mid s_{0:t-1}, y_{0:t}) = p_{\Theta'}(s_t \mid s_{0:t-1}, y_{0:t})$ which is the distribution that minimizes the variance of the importance weights given $s_{0:t-1}$ and $y_{0:t}$ [14]. In addition, it is easy to obtain samples from $p_{\Theta'}(s_t \mid s_{0:t-1}, y_{0:t})$ as it can be computed using κ Kalman filters

$$p_{\Theta'}(s_t \mid s_{0:t-1}, y_{0:t}) = \frac{p_{\Theta'}(s_t \mid s_{t-1}) \cdot p_{\Theta'}(y_t \mid s_{0:t}, y_{0:t-1})}{p_{\Theta'}(y_t \mid s_{0:t-1}, y_{0:t-1})}, \tag{34}$$

where

$$p_{\Theta'}(y_t \mid s_{0:t-1}, y_{0:t-1}) = \sum_{s_t=1}^{\kappa} p_{\Theta'}(s_t \mid s_{t-1}) \cdot p_{\Theta'}(y_t \mid s_{0:t}, y_{0:t-1}). \tag{35}$$

The prior distribution $p_{\Theta'}(s_t|s_{t-1})$ can also be used as an instrumental distribution. In this case, based on (33), the computation of importance weights requires only one step of a Kalman filter, but the variance of the importance weights is large.

The k-th iteration of the proposed method for estimation the parameters using a sequential Monte Carlo (MC) approximation of the EM algorithm is presented in Algorithm 1. Its computational complexity is equal to $\mathcal{O}(N\,T)$, where N is the number of particles and T is the size of the learning dataset.

4.2.2. Approximation of the marginal likelihood

The marginal likelihood $p_\Theta(y_{0:T})$ is approximated by

$$\widehat{p}_\Theta(y_{0:T}) = p_\Theta(y_0) \prod_{t=1}^{T} \widehat{p}_\Theta(y_t \mid y_{0:t-1}), \tag{36}$$

where $\widehat{p}_\Theta(y_t \mid y_{1:t-1})$ is iteratively calculated using the Kalman filter and the importance weights

$$\widehat{p}_\Theta(y_t \mid y_{0:t-1}) = \sum_{i=1}^{N} \hat{w}_{t-1}^i \widehat{p}_\Theta(y_t \mid s_{0:t-1}^i, y_{0:t-1}). \tag{37}$$

Algorithm 1 k-th iteration of the MC-EM algorithm for SMSSM

Input $\leftarrow \Theta = \Theta^{(k)}, y_{0:T}, u_{0:T}$
Init $\leftarrow s_0^i \overset{i.i.d.}{\sim} \Pi \ (\forall i = 1:N)$
for $t = 1:T$ and $i = 1:N$ **do**
1. Sample $s_t^i \overset{i.i.d.}{\sim} p_\Theta(s_t \mid s_{0:t-1}^i, y_{1:t})$, thus $s_{0:t}^i = \left(s_{0:t-1}^i, s_t^i\right)$
2. Given $s_{0:t}^i$ and $y_{1:t}$, calculate x_t^i using a Kalman filter
3. Calculate importance weights w_t^i
4. Selection step: sample $s_{0:t}^i \overset{i.i.d.}{\sim} \sum_{i=1}^{N} w_t^i \delta(s_{0:t} - s_{0:t}^i)$, where $\delta(\cdot)$ is a Dirac function with mass at zero

end for
for $t = 1:T$ and $i = 1:N$ **do**
Calculate $\hat{x}_{t|T}^i$ and $\hat{x}_{t,t|T}^i$ using a Kalman filter
end for
Estimate $\Theta^{(k+1)}$ by solving $\partial\mathcal{L}(\theta, \lambda, \mu)/\partial\Theta = 0$ using (29)

In this section, we have studied the inference of the unknown parameters Θ assuming that the number of the hidden Markov states κ is fixed. In the next section, we are interested in the choice of the optimal κ.

5. Order Selection for SMSSM

The hidden Markov state models the different regimes of the underlying dynamics of the system. However, no hints are available with regards to the number of these regimes. As a consequence, the number of hidden Markov states κ should be identified. The first issue to be discussed is the identification criterion of the optimal κ. This criterion is commonly determined by a "trade-off" between the accuracy requirements and the model complexity. The accuracy of the model is determined by its capacity to fit the data. In this paper, the likelihood is used as an accuracy indicator. This is justified

by the fact that the model is inferred using the ML approach. However, the maximum of likelihood does not decrease with κ, and should be penalized by the complexity of the model. This complexity depends essentially on the number of independent parameters K. For a SMSSM, K is equal to $\kappa^2 + (2n+3)\kappa$. The leading penalized likelihood criterion are BIC [15] and AIC [16]. Both are derived from distinct perspectives and address differently the trade-off between accuracy and complexity. Indeed, BIC is "consistent" in the sense of asymptotically selecting the true model (i.e., the probability of selecting the true model by BIC approaches to one as $T \to \infty$), whereas AIC is "efficient" in the sense of asymptotically minimizing the mean-squared prediction error (see for instance [17]).

In practice, the numbers minimal and maximal of the hidden Markov states are fixed. The maximum of likelihood of all possible κ is then calculated using the presented MC-EM algorithm. Finally, the optimal κ with respect to BIC (resp. AIC) is the one that minimizes BIC (resp. AIC). Algorithm 2 presents the selection of the optimal κ using BIC and AIC.

Algorithm 2 Selection of the optimal κ using BIC and AIC

for $\kappa = \kappa_{\min} : \kappa_{\max}$ **do**

1. Calculate $\hat{p}_\Theta(y_{0:T})$ from (36)
2. $BIC(\kappa) = -2 \log \hat{p}_\Theta(y_{0:T}) + K \log(T)$
3. $AIC(\kappa) = -2 \log \hat{p}_\Theta(y_{0:T}) + 2\,K$

end for

Output $\leftarrow$ $\kappa^\star_{BIC} = \arg\min_\kappa BIC(\kappa)$

Output $\leftarrow$ $\kappa^\star_{AIC} = \arg\min_\kappa AIC(\kappa)$

Given the number of hidden Markov states and the corresponding model parameters Θ, the online estimation of the state of interest x_t is considered in the next section.

6. Online estimation of the state of interest

We suppose here that the number of hidden Markov states has been previously identified (see Section 5) and that the associated vector of parameters Θ is estimated (see Section 4). The optimal estimations of x_t and s_t is a well-known NP problem and requires a prohibitive computational cost. Indeed, a closed form solution would require running Kalman filters for each possible Markov sequence $s_{0:t}$; i.e., κ^{t+1} filters. To overcome this computational

problem, a variety of suboptimal estimation algorithms has already been proposed. Most of these algorithms are based on merging the components at time t like the Interacting multiple model (IMM) [18]. The IMM uses only κ parallel Kalman filters and then merges their κ estimated x_t^i using a deterministic finite Gaussian mixture approximation. Another possible suboptimal strategy is to approximate the continuous process X_t by a finite state process with fixed states. The SMSSM is thus reduced to a HMM with two discrete latent states, and the estimation algorithms of a HMM can be easily implemented. A more sophisticated method uses particle filter to estimate x_t. As shown in Section 7, N sequences of Markov states $s_{0:t}^i$ are sequentially simulated from an instrumental distribution $q_\Theta(s_t|s_{0:t-1}^i, y_{0:t})$, then $x_t^1, \ldots, x_t^N$ are obtained using the Kalman filter. Finally, x_t is estimated either by the empirical mean

$$\widehat{x}_t = \frac{1}{N}\sum_{i=1}^{N} x_t^i, \tag{38}$$

or by a confidence interval

$$\widehat{x}_t \in [x_t^{0.05}; x_t^{0.95}], \tag{39}$$

where x_t^p denotes the empirical p-quantiles.

In the next section, the issues that arise in implementing SMSSMs are analyzed, namely the influence of the number of particles on both off-line inference of unknown parameters and online estimation of x_t, and the initialization strategy of the MC-EM algorithm.

7. Sensitivity Analysis

In this section, we analyze the numerical difficulties in implementing SMSSMs, and validate the presented MC-EM algorithm as well as the online estimation of x_t. We also verify the ability of the model selection criteria, AIC and BIC, to identify the optimal model order for a SMSSM. To this purpose, we simulate a learning dataset with $T = 500$ through a SMSSM with $\kappa = 3$ (Fig. 1). Table 1 summarizes the values of its parameters. The main difficulty in this model is that C is well below 1, with C describing the influence of the state of interest X_t on the observation Y_t. When C has small values, meaning that X_t has a very limited impact on Y_t, it is then difficult to estimate X_t based solely on Y_t. It is noteworthy that this simulated data

	A	$B = [B_1\ B_2]$	C	$D = [D_1\ D_2]$	σ_X^2	σ_Y^2
$s = 1$	1	[1.20 0]	0.11	[350 375]	10^{-4}	20
$s = 2$	1	[1.35 0]	0.15	[400 380]	10^{-4}	1
$s = 3$	1	[1.30 0]	0.20	[420 385]	10^{-4}	1

Table 1: Parameters values of the SMSSM with $\kappa = 3$ used to create the simulated dataset

matches the behavior of a battery. In addition, similarly to the battery case, we consider that A and B_2 are identified in advance.

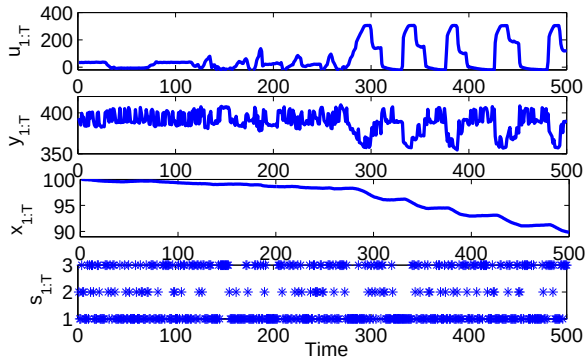

Figure 1: [Simulated data] Learning dataset simulated through a SMSSM with $\kappa = 3$

7.1. MC-EM algorithm

The accuracy of the presented parameters inference algorithm depends, on the one hand, on the initialization of the set of unknown parameters as the EM may converge to a local maximum of likelihood and, on the other hand, on the number of particles since the EM algorithm is approximated using a Monte Carlo method.

7.1.1. Initialization strategy

Independent initialization. Each unknown parameter is initialized independently through a Gaussian distribution with mean computed based on its physical interpretation and/or expert knowledge. The variance should be relatively large to ensure that the initialization varies each time the MC-EM algorithm is run. The variance of the process noise ω_t and measurement noise ε_t are set to 10^3, as the transition and observation equations are unreliable at this stage. In addition, the transition matrix P is initialized by assigning

random values to each row which is then normalized. Figure 2 shows the results of the estimated ML for $\kappa = \{2, \ldots, 7\}$ based on the simulated data. It can be observed that some estimations may be far from being optimal. Contrary to the results obtained for $\kappa = 5$ and 7, the maximum of likelihood must not decrease with κ. In order to overcome this problem, a different initialization strategy, called "nested initialization", is used [19].

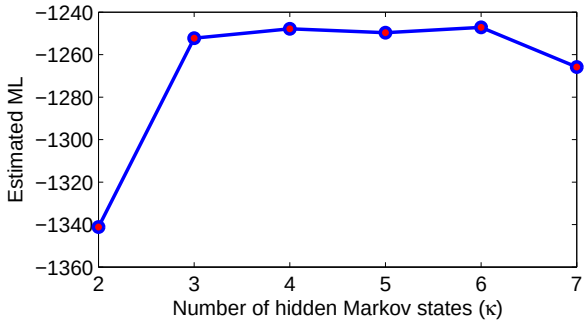

Figure 2: [Simulated data] Estimated ML for $\kappa = \{2, \ldots, 7\}$; 20 independent initializations of the EM algorithm for each κ

Nested initialization. called also $Km1$ for "K minus 1". For $\kappa \geq 2$, suppose that Θ of model order $\kappa - 1$, denoted $\Theta_{(\kappa-1)}$, is available. Then, to initialize $\Theta_{(\kappa)}$, the state κ_0 having the maximum entropy is divided into two states. Figure 3 illustrates this strategy. The entropy of a state r is computed as follows

$$\mathrm{E}(r) = -\sum_{t=0}^{T} p_\Theta(s_t = r \mid y_{0:T}) \log p_\Theta(s_t = r \mid y_{0:T}), \tag{40}$$

where $r = arg\max_s \; p_\Theta(s_t = s \mid y_{0:T})$. In Figure 4 the parameters vector

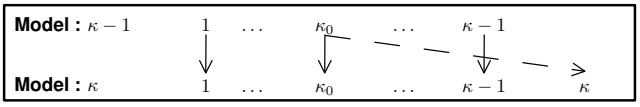

Figure 3: Nested initialization strategy ($Km1$ strategy) for the EM algorithm

$\Theta_{(2)}$ is estimated based on the simulated data. To initialize the parameters for $\kappa = 3$, the entropy of the two states are computed. Figure 4 shows that the observations under the state having the highest entropy (*e.g.* state 1)

are more dispersed than those under the other states (*e.g.* state 2). Thus for $\kappa = 3$, the state 1 is divided into two states and the parameters vector of state 1 and 3 is initialized as follows

$$\Theta_{(3)} = \{\Theta_{(2)}(s=1) + random_1,\ \Theta_{(2)}(s=2),\ \Theta_{(2)}(s=1) + random_2\}.$$

In Figure 5, it can be observed that the problem of suboptimal ML solutions

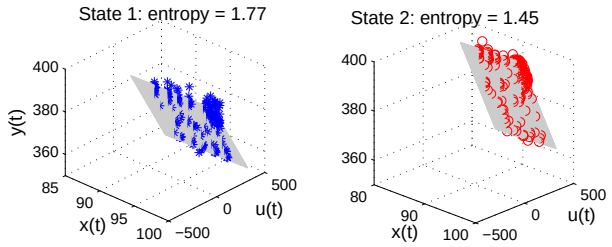

Figure 4: [Simulated data] Distribution of the observations around the estimated parameters per state: the observations under the State 1 are more dispersed than those under the State 2. State 1 should be divided into two modes

is resolved, and the estimated MLs with nested initializations increase with respect to the model order κ.

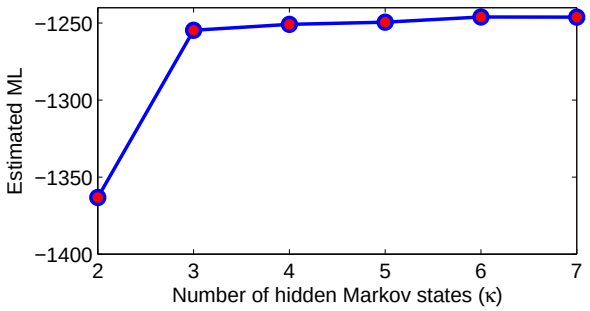

Figure 5: [Simulated data] Estimated ML with $\kappa = \{2, \ldots, 7\}$; 20 nested initializations of the EM algorithm for each κ

7.1.2. Influence of the number of particles

In order to study the influence of the number of particles on the estimated ML, we run 20 EM with identical initialization for $N = 100$ and $N = 1000$. Figure 6 shows that the mean of the estimated ML is almost equal in both

cases. However, the dispersion of the estimated ML for $N = 100$ is relatively higher than the one for $N = 1000$. Figure 7 shows the estimated MLs for

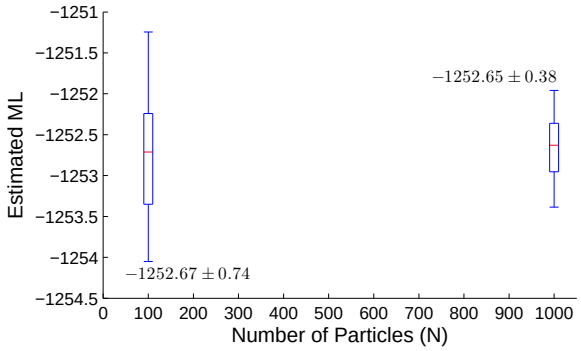

Figure 6: [Simulated data] Estimated ML for $N = 100$ and 1000 with $\kappa = 3$; 20 identical initializations of the EM algorithm for each case

different number of particles N with $\kappa = 5$. For each N, the EM algorithm is repeated 20 times with different independent initializations. For each N the dispersion of the estimated MLs has two sources: the initialization of the EM algorithm and the Particle filter. We note that, in almost all cases, this dispersion remains relatively small. Hence, the choice of the likelihood as an accuracy indicator of a SMSSM (Section 5) seems pertinent.

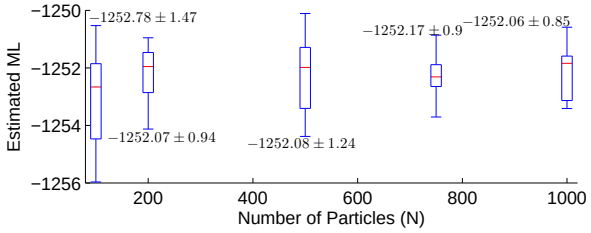

Figure 7: [Simulated data] Estimated ML for $N = \{100, 200, 500, 750, 1000\}$ with $\kappa = 5$; 20 different initializations of the EM algorithm for each case

7.1.3. Validation of the parameters estimation methods

In this section, we compare the true parameters of the simulated model with $\kappa = 3$ and estimated parameters for $\kappa = 3$ and 5. For $\kappa = 3$, Figure 8 highlights that $B(s)$ the parameter of the equation of the unobserved state X_t is less well-estimated than the parameters of the equation of the observed

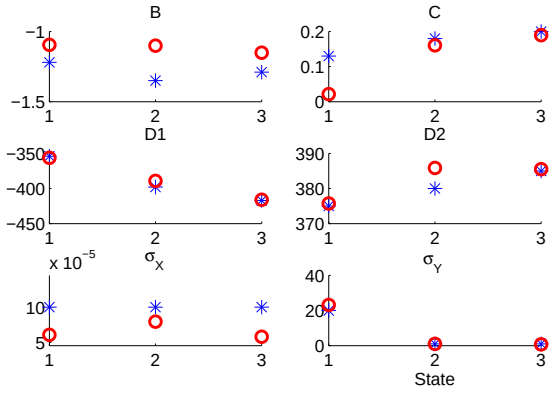

Figure 8: [Simulated data] Comparaison between true ($*$) and estimated ($\circ$) parameters with $\kappa = 3$

variable Y_t. For $\kappa = 5$, Figure 9 highlights that State 1 is duplicated into 2 states: State 3 and 5. In this case, for $\kappa > 3$, the EM algorithm duplicates a state of the true model $\kappa = 3$ in order to find the set of parameters that maximizes the likelihood $p_\Theta(y_{0:T})$. This can be clearly seen in Fig. 5 where the estimated likelihood is almost stable for $\kappa \geq 3$.

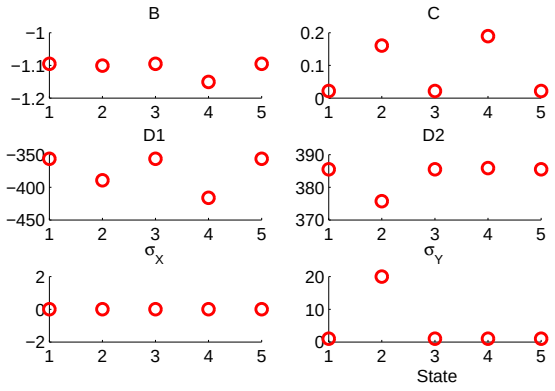

Figure 9: [Simulated data] Estimated parameters for SMSSM with $\kappa = 5$

7.2. Model selection using BIC and AIC

Based on the simulated data, BIC and AIC find the true number of Markov states ($\kappa = 3$), cf. Figure 10. By examining each criterion closely, we

note that BIC chooses sharply $\kappa = 3$, however AIC hesitates between $\kappa = 3$ and 4. This is because AIC tends to overestimate the model order (see for instance [20]). This will emerge clearly when using real data.

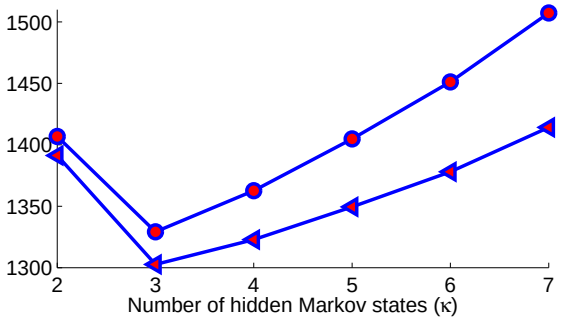

Figure 10: [Simulated data] BIC ($\circ$) and AIC ($\triangle$) with $\kappa = \{2, \ldots, 7\}$

7.3. Online estimation of x_t

The number of particles N to be used for the online estimation of x_t is constrained by the limited hardware ressources in embedded applications. Based on the simulated data, the results show that a small number of particles ($N \simeq 10$) is sufficient to accurately estimate x_t (Fig. 11 and 12).

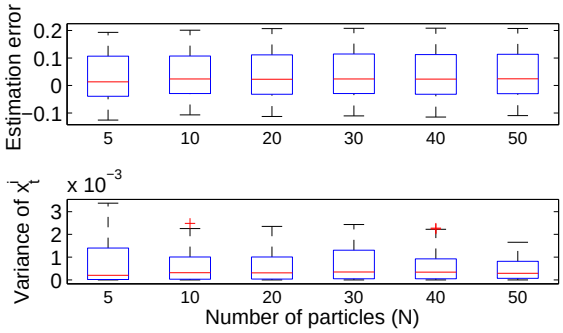

Figure 11: [Simulated data] Influence of the number of particles N on the online estimation of x_t

In the next section, we implement a SMSSM in real-world application, namely the estimation of the state of charge of an electric battery.

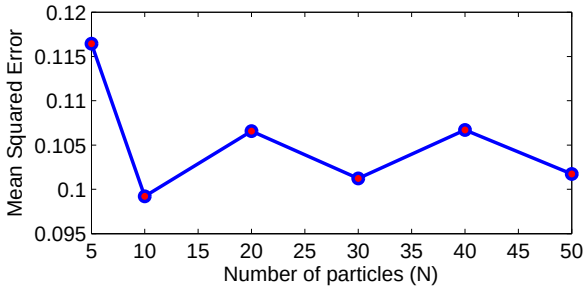

Figure 12: [Simulated data] Mean squared error of the online estimated x_t vs. the number of particles N

8. Estimation of the State of Charge of an electric battery using a SMSSM

Our study is motivated primarily by the problem of estimating the State of Charge (SoC) of battery for electric vehicles. Indeed, the battery dynamics randomly changes according to uncontrolled usage conditions such as ambient temperature and driving behavior. An accurate SoC of a battery is essential to indicate its autonomy and its available energy, as well as to guarantee a safe utilization of the battery by preventing under or over-charge that may lead to permanent damage.

8.1. State of art

A review of the SoC estimators as well as their performances in embedded applications is given in [21]. The most commonly used approaches are founded on the "Coulomb counting" model and on the general state-space models. The Coulomb counting establishes a SoC model based on a weighted summation of input and output battery currents. Although simple, this method is an open loop. As a result, the error of the current sensor can drift the estimation. In order for it to be reliable, this method requires an accurate, thus expensive, current sensor. On the other hand, a general state-space model combines the SoC, modeled using Coulomb counting, and the voltage, modeled using an equivalent electric circuit. As a result, recursive SoC estimation can be provided by an extended Kalman filter. It is noteworthy that, a Kalman filter being a closed loop method, it can take the sensor error into account. However, real-life conditions can lead the state-space model describing the behavior of an electric vehicle battery to change

over time. Indeed, these changes cannot be predicted, as they are influenced by uncontrolled external conditions, such as the ambient temperature and current profile, but also by internal conditions like the battery resistance and aging. Attempts have been made to improve the Kalman filter method so as to integrate the possibility of changes over time. In [4], sets of parameters are identified for several temperatures and SoC. In [5], unknown parameters are included in the state vector, and then estimated at each time t. These solutions remain contestable. Indeed, in the former the parameters changes according to an estimated, and possibly inaccurate, SoC. The latter is too expensive and thereby not suitable for an online application. Furthermore, changes can be modeled through a regression function relating each parameter with a given temperature as in [6]. However, this method ignores the influence of uncontrolled internal and external conditions. To the best of our knowledge, we can stipulate that there is no model or method that gives an accurate online SoC regardless of internal and external usage conditions.

8.2. Description of the learning dataset

The learning dataset (see an example Fig. 13) gathers current and voltage measurements collected during drive of an EV, with an ambient temperature of $15°C$, a sampling time of 1s and a working time of 4500s. The SoC was computed using the Coulomb counting method as the EV was equipped with an accurate current sensor.

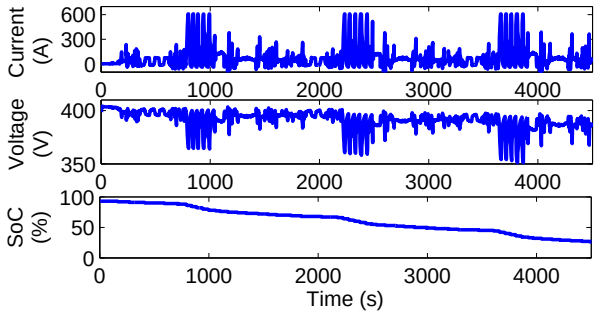

Figure 13: [Real data] Learning dataset collected during the drive of an Electric Vehicle

8.3. Motivations

Our aim is to develop a generic SoC estimator able to take into account the random variations of the battery dynamics. At first, we attempt to

estimate the SoC using a LGSSM (i.e., SMSSM with $\kappa = 1$). In order to monitor the relevance of the Kalman filter, attention has been paid to compare the estimated voltage with the observed voltage. Figure 14 shows that a single LGSSM cannot estimate accurately the voltage throughout the whole interval $[0\,;\,T]$, and different potential LGSSMs should be used instead.

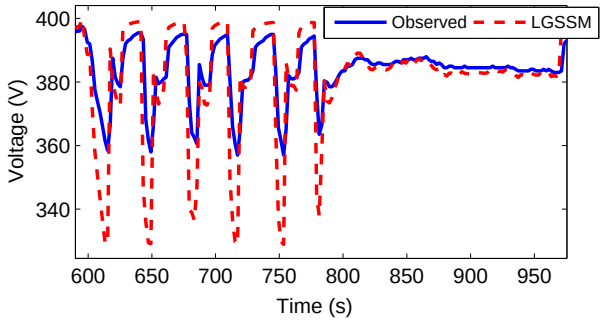

Figure 14: [Real data] Prediction of the voltage using a linear Gaussian state-space model

8.4. SoC model

By now the SoC is modeled by the following SMSSM, for which the observation and the state equations are based on physical models [6]

$$\begin{aligned} X_t &= X_{t-1} + B(S_t)u_t + \omega_t, \\ Y_t &= C(S_t)x_t + D_1(S_t)u_t + D_2(S_t) + \varepsilon_t, \end{aligned} \tag{41}$$

where Y_t is the observed voltage, u_t the input current and X_t the SoC to be estimated. Here, $A(s)$ is physically identified: $\forall s$, $A(s) = 1$. In addition, we have a physical prior information at $t_0 = 0$

$$y_0 = C(s_0)x_0 + D_2(s_0), \tag{42}$$

where y_0 is the Open Circuit Voltage (OCV) measurement and x_0 its corresponding SoC. Indeed, in practice at $t_0 = 0$, the battery is often in a resting state, and the SoC can be efficiently calculated using an OCV/SoC relationship. To validate our SoC model, as well as the parameters estimation method, real-life Electric Vehicle (EV) data are used. Accordingly, the

battery usage does not only depend on ambient temperature and itinerary, but also on drivers behavior and road conditions.

8.5. Choice of the optimal number of Markov states

The number of hidden Markov states κ reflects the number of different regimes of the underlying dynamics of the battery. Figure 15 shows that the optimal κ is equal to $\kappa = 4$ with respect to BIC, and $\kappa = 5$ with respect to AIC. We note that AIC gives close results for $\kappa = 4$, 5 and 6, but that BIC chooses sharply $\kappa = 4$. The model chosen by BIC is more parsimonious and requires less storage and computation capacities. This makes it appropriate for electric vehicles as it satisfies embedded application constraints. On the other hand, the model chosen by AIC theoretically minimizes the estimation error of the SoC. Thus, depending on given selection criteria, such as the estimation error or the computation time, the "optimal" number of hidden Markov states can be chosen.

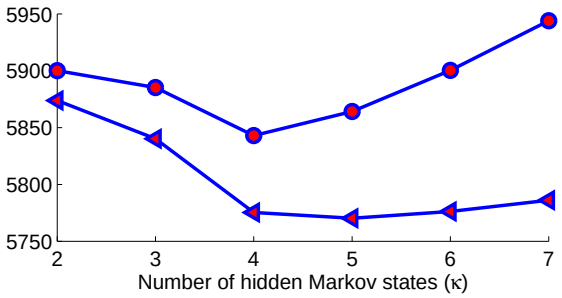

Figure 15: [Real data] BIC ($\circ$) and AIC ($\triangle$) with $\kappa = \{2, \ldots, 7\}$

8.6. Validation of the model

The learned SMSSM with $\kappa = 4$ (and an ambient temperature of $15°C$) is validated using three different datasets also collected during a drive of an EV, under different ambient temperatures (5, 15, $25°C$). The results show that SMSSM provides an accurate and robust SoC estimation even for different ambiant temperatures. Indeed, the maximum difference between the SoC estimated by Coulomb counting and SMSSM is equal to 5%; whereas this difference reaches 20% for a single LSSM, Fig. 16. Moreover, numerical experiments show that under a specific hidden Markov state, the relation between SoC, voltage and current is linear (Fig. 17) which confirms the

SMSSM assumptions. Figure 18 shows that the hidden Markov state could reflect a specific usage of the battery as it follows closely the variation of the voltage. Thus, we expect that this hidden state would have a physical interpretation and would model the physical changes of the battery behavior.

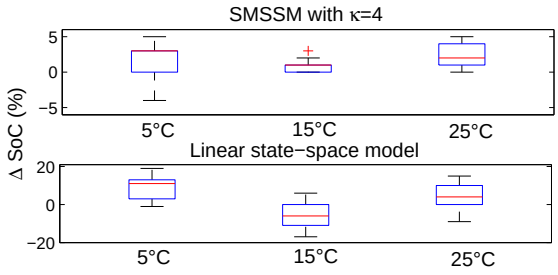

Figure 16: [Real data] Difference between Coulomb counting and estimated SoC using SMSSM with $\kappa = 4$ (top) and Linear state-space model (bottom) for ambient temperature equal to 5, 15, 25°C

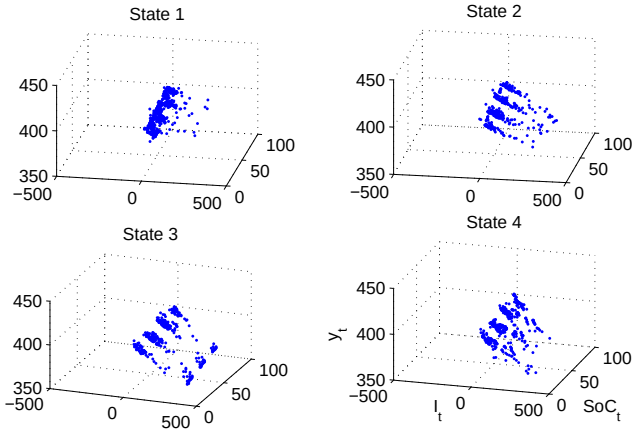

Figure 17: [Real data] Voltage vs SoC vs current: instantaneous measurements under each hidden Markov state for a SMSSM with $\kappa = 4$

9. Discussion

This paper has addressed the four basic problems in implementing a switching Markov state-space model in real-world applications, namely the

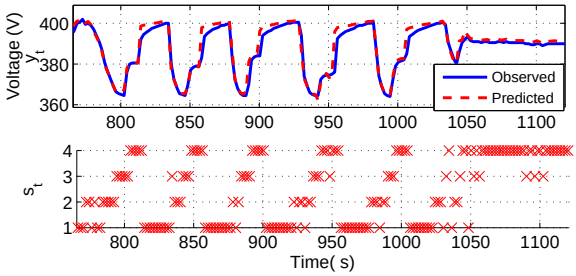

Figure 18: [Real data] Variation of the voltage and the hidden Markov state over time. The presented s_t is the one that appears the most in the N particles s_t^i

identifiability, the off-line inference of the model parameters, the order selection and the state of interest inference. For the identifiability issue, we have proven that, in case of an irreducible and aperiodic Markov chain, a constraint relating the state of interest X_t to the observations Y_t at a time t_0 is sufficient to ensure the identifiability of the model parameters. Under these constraints, an off-line inference of these parameters is then developed using a Monte Carlo approximation of the EM algorithm. The numerical difficulties in implementing this algorithm have been studied based on a simulated dataset. The results show that a nested initialization of the EM algorithm is more appropriate than an independent one in case of SMSSMs. Indeed, the nested initialization takes into account the characteristics of mixture models, and ensures that the transition from order κ to $\kappa + 1$ improves the model (i.e., the estimated maximum of likelihood does not decreases). Another numerical issue lies with the choice of the number of particles. The results for $N = \{100, 200, 500, 750, 100\}$ shows that the dispersion of the estimated ML is relatively small. This confirms the use of penalized likelihood criteria, namely AIC and BIC, to select the number of hidden Markov states κ. Once the number of hidden Markov states is chosen and the corresponding set of model parameters is estimated, they can be embedded into the system management controller to provide an online estimation of the state of interest X_t. This paper has presented an online state inference using a particle filter. The sensitivity analysis has shown that the required number of particles is $N \simeq 10$, rendering the use of the particle filter feasible in embedded applications. In order to illustrate a real-world application of SMSSMs, the problem of estimating the SoC of an electric battery, using a SMSSM-based model, is considered. Indeed, the hidden Markov states of this SoC model reflect the

random variations of the dynamics of the battery induced by uncontrolled usage conditions. Moreover, the constraints ensuring the identifiability of this *SoC* model have been physically identified. Finally, the validation of this model through real-world electric vehicle data confirmed its ability to accurately estimate the *SoC* under different driving and ambient temperature conditions. An alternative method for estimating the model parameters, based on Gibbs sampler, is being explored.

Acknowledgment

The authors would like to thank Michelin for their contribution on CEA pack integration into Michelin vehicle and for performing tests on Michelin circuit. The authors would also like to thank Gilles Celeux for his valuable comments.

References

[1] S.M. Oh, J.M. Rehg, T. Balch, and F. Dellaert, "Learning and inferring motion patterns using parametric segmental switching linear dynamic systems", *International Journal of Computer Vision*, vol. 77, no. 1-3, pp. 103-124, 2008.

[2] M. Lammerding, P. Stephan, M. Trede, and B. Wilfling, "Speculative bubbles in recent oil price dynamics: evidence from a Bayesian Markov-switching state-space approach", *Energy Economics*, vol. 36, pp. 491-502, 2013.

[3] S. Frühwirth-Schnatter, "Finite Mixture and Markov Switching Models", *Springer*, 2006.

[4] T. Hirai, A. Ohnishi, N. Nagaoka, N. Mori, A. Ametani, and S. Umeda, "Automatic equivalent-circuit estimation system for lithium-ion battery," *Universities Power Engineering Conference, 43rd International*, pp. 1-5, 2008.

[5] S. Pang, J. Farrell, J. Du, and M. Barth, "Battery state-of-charge estimation", *American Control Conference*, vol. 2, pp. 1644-1649, 2001.

[6] L.G. Plett, "Extended Kalman filtering for battery management systems of LiPB-based HEV battery packs", *Journal of Power Sources*, vol. 134, pp. 262-276, 2004.

[7] E. Walter and Y. Lecourtier, "Unidentifiable compartmental models: what to do?", *Mathematical Biosciences*, vol. 56, no. 1-2, pp. 1-25, 1981.

[8] M. Stephens, "Dealing with label switching in mixture models", *Journal of the Royal Statistical Society, Series B*, vol. 62, pp. 795-809, 2000.

[9] S.J. Yakowitz and J.D. Spragins, "On the identifiability of finite mixtures", *The Annals of Mathematical Statistics*, vol. 39, pp. 209–214, Feb 1968.

[10] B.G. Leroux, "Maximum-likelihood estimation for hidden Markov models", *Stochastic Processes and their Applications*, vol. 40, no. 1, pp. 127-143, 1992.

[11] A.P. Dempster, N.M. Laird, and D.B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm", *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 1–38, 1977.

[12] A. Doucet, N. De Freitas, and N. Gordon, "Sequential Monte Carlo methods in practice", *Springer-Verlag*, 2001.

[13] N.J. Gordon, D.J. Salmond and A.F.M. Smith, "Novel approach to nonlinear/non-Gaussian Bayesian state estimation", *Radar and Signal Processing, IEE Proceedings F*, vol. 140, no. 2, pp. 107-113, 1993.

[14] A. Doucet, N. Gordon, and V. Krishnamurthy, "Particle filters for state estimation of jump Markov linear systems", *IEEE Transactions on Signal Processing*, vol. 49, pp. 613-624, Mar 2001.

[15] G. Schwarz, "Estimating the dimension of a model", *The Annals of Statistics*, vol. 6, no. 2, pp. 461–464, 1978.

[16] H. Akaike, "Information theory and an extension of the maximum likelihood principle", *Selected Papers of Hirotugu Akaike*, pp. 199–213, Springer, 1998.

[17] K. Burnham and D. Anderson, "Model selection and multimodel inference: a practical information-theoretic approach", *Springer*, 2002.

[18] H.A.P. Blom and Y. Bar-Shalom, "The interacting multiple model algorithm for systems with Markovian switching coefficients", *IEEE Transactions on Automatic Control*, vol. 33, no. 8, pp. 780-783, 1988.

[19] J.-P. Baudry and G. Celeux, "EM for mixtures - Initialization requires special care", *Statistics and Computing*, 2015 (to appear, https://hal.inria.fr/hal-01113242).

[20] G. Celeux and J. Durand, "Selecting Hidden Markov Model state number with cross-validated likelihood", *Computational Statistics*, vol. 23, no. 4, pp. 541–564, 2008.

[21] J. Kalawoun, K. Biletska, F. Suard, and M. Montaru, "From a novel classification of the battery state of charge estimators toward a conception of an ideal one", *Journal of Power Sources*, vol. 279, pp. 694-706, 2015.